

Jack Mc Donald  
745 - 7 rue D. Ballard

☒ 1455 de Maisonneuve Blvd. West  
Montreal, Quebec H3G 1M8

☐ 7141 Sherbrooke Street West  
Montreal, Quebec H4B 1R6

SEP 21 1979



Tel. (514) 879-4501

Dept. Of Education H 549

17 September, 1979

Carolyn Stauffer  
Apple Computer Inc.  
10260 Bandley Drive  
Cupertino Calif. 95014

Dear Ms. Stauffer,

Perhaps you recall our conversation at ADCIS San Diego concerning the relationship needed to encourage educators to author good learning activity software?

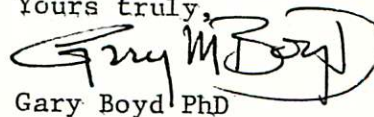
In any case, I did enjoy chatting with you there. Subsequently, I have been called upon to produce some working papers for the Canadian National Research Council on courseware dissemination. In the course of preparing these I developed a more general scheme for marketing and recovering royalties (which I have written about in an article for Computer next spring).

It occurs to me that you might be interested in this scheme and the underlying perspective; hence, I enclose a copy of the article.

We have recently acquired an Apple II for use by our graduate students in Educational Technology, and I would appreciate information on any education and/or training software and/or courseware which you have available currently.

Also, it is possible that some of the materials which our grad students are producing and testing may be of interest to you in the future.

Yours truly,

  
Gary Boyd PhD

Assoc. Prof. Ed. Tech.

incorporating Sir George Williams University and Loyola of Montreal

Go-Between-Batteries ;

Secure Rechargeable, Interpreter / Auditor

Modules as a Means for Marketing

Educational and Other Dialogware

by: Gary M. Boyd

Concordia University of Montreal

1. The Problem A problem is a "sought-after relation" according to Gordon Pask. The relation sought here is one which will permit students, teachers, and others easy access to repair-able, extendable dialog programs, while at the same time ensuring authors of receiving audited usage reports and evaluative data, and hence the possibility of collecting royalties proportional to the use made and the quality of their contributions.

If one sells interactive applications packages (e.g. CAI courseware) in the form of object-code cassettes or diskettes etc., it frequently turns out to be : a) unreliable (drop-outs, noise, etc.) b) un-repairable by the user c) impossible to modify to adapt to local needs d) altogether too easy to copy and pass on without either paying royalties or even acknowledging the source. On the other hand, if one sells source code for a well-known language (BASIC, TUTOR, etc.), the package will be easy to fix, and to adapt to local needs, but it will also be more tempting to give away or to steal. Licensing arrangements are not effective when myriad small users are involved, and most large systems have security

problems. There seems to be no really good way to sell packaged dialogs and other interactive applications packages.

In order to market CGU (Computer Guided User) and CAI (Computer Assisted Instruction) dialogs which will be repair-able, and modifiable, and will yield needed returns to their authors, a new approach is required.

The valuable thing to the user is the quantity of high-quality interaction with the dialog program; that is, the time spent productively interacting with potent, timely, and valuable routines. The valuable things to the author are either royalty fees, or an audited statement-of-use which can be employed to get grants and other resources in order to produce more programs, or to do more research. One would like to know how much the program is used and how highly the user rated the portions of it as he or she used them.

In short, the problem is to engineer an accomodation between user desires and author/producer desires.

## 2. The Proposed Solution: "Go- between Batteries"

The proposed solution is that scaled packages containing an interpreter and usage recorder and encryption and decryption components, which communicates by some standard interface such as an IEE 488 buss, be leased to customers



on a charge-for-the-quantity-of-use basis similar to that employed for postage meters. Each time the "Go-between Interpreter / Auditor battery" is connected to the agency's network for re-charging, its memory, storing the code names of all the dialog programs which have been interpreted for its users, will be read out by the auditing agency. This audited information will then be passed back to the respective dialogware authors, together with the appropriate royalties.

Educational Courseware and other dialogware authors would supply a copy of their source code together with the program name and an identifier statement, which would be an important short statement and statement number, from somewhere in the program, encrypted according to the public-key of the marketing agency.

When the programs are 'played' by users, the "Go-be batts" will decrypt the identifier statement and number, and check to see if the actual source code being interpreted has the corresponding statement in the right place. The outcome of this check, and the code name of the dialog program will both be stored in the "Go-be batt" for reporting back at the next recharge.

### 3. The Rationale for this Solution

Compact sophisticated interpreters for high-level dialog languages capable of being employed in conjunction with small (eg. personal) computers are and will probably remain very difficult and expensive to develop. It does not matter if, or how, people copy the source code for dialogs if the only way they can run these is through a special interpreter which

records and reports back what it has interpreted.

Providing large quantities of one type of standard sealed 'rechargeable' interpreter module is a much more economic approach than providing thousands of different sealed courseware cartridges. The provision of dialogware as object-code is a poor expedient since tiny errors in duplication lead to un-repairable bugs. Moreover, the content and approach is hidden so that the buyer is faced with the 'pig in a poke' situation. ,

#### 4. Some Objections, and Their Circumvention

An obvious objection is that if the specifications for the high-level dialog language are public enough for people to write and modify their own program, then it will be possible for counterfeit interpreters to be produced. This is true, but the cost of developing sophisticated compact interpreters ( or operating systems) and the risks due to rights infringements involved in distributing such devices would not be justified by the probable returns.

The sophistication of the dialog language and hence its interpreter is a crucial variable to this approach. If the dialog language is too simple, like PILOT for example, then many people can write their own 'home-brew' interpreters and the reporting/royalty scheme is defeated. If, on the other hand, the language is too sophisticated, ordinary users will not be able to debug source code nor will they be able to add their own contributions.

Also, if the language is too sophisticated, even the essential core of an interpreter will be too big to fit into a sealed cartridge. Probably the best current CGU/CAL languages are TUTOR and NATAL; they exist only on large computers at present. (PLATO systems, / and PDP 10 or IBM 370 systems (NATAL).) Their interpreters require in the neighborhood of 50K to 100K bytes of fast memory. Such interpreters could, however, be partitioned so that only what is needed for a given section of a dialog would be held in fast memory during execution. Currently, a mini-computer implementation of NATAL is being produced for the Canadian National Research Council, based on this interpreter partitioning approach. The full formal specifications of both these languages are public, and hence, neither could be used 'as is' to form part of a secure "Go-between battery" marketing system; however, they are good examples of languages with powerful conversational capabilities, both alphanumeric and graphic, which are almost ideal for computer assisted learning, and computer guided user applications.

This same approach could be used for providing the main part of a proprietary operating system which in a sense is also go-between software.

Although "Go-be batts" are referred to, at first the sealed packages would probably contain standard micro-processors and memories; only when the market volume gets up will actual LSI firmware interpreters be designed and manufactured.



The readout and recharging operations have to be secure. The read-in of program names does not need to be. If I write a program and add someone else's valid program name to it in order to run it via the interpreter, or if I add improvements to someone else's program, I will simply be giving any royalties or plaudits which accrue to the owner of the valid code name!

The security of recharging requires that the package self-destruct if an attempt is made to open it;

either a software code or a peculiar connection configuration or a combination of both is needed. In the not too distant future, Rivest-type encryption and decryption LSIs should be cheap mass-produced items and perhaps one of these could be used as a gatekeeper.

Whether "Go-be batts" should be designed to work with second generation personal computers or with teletext (in Canada, TELIDON) home TV communication processors, is as yet an open question. The advantage of TELIDON is its superior color graphics, but personal computers (such as Apple II etc.) can have more local storage and processing capability at least at present.

Overhead; another objection is that the system is too complicated and that it is simpler just to sell schools

turn-key systems which accept only the manufacturers' encrypted dialogware. This solution is very disadvantageous to users since they cannot easily participate in modifying, improving, or authoring their own dialogware. One of the main advantages of "Go-be batts" is that the recharging agency is in effect a public market through which anyone and everyone can sell and buy access to programs.

### 5. Quality Control

If the size of royalty is based not only on the usage time but also on the quality of the interaction, how can one elicit honest evaluations of quality if these will drive the price up for the user when they are favorable? If royalties are based only on the interaction-time used, then there would be a tendency to write slow and long-winded programs. More seriously, consumers would have no a-priori way of assessing probable program value independent of the suppliers' advertisements.

The solution to this problem lies in the leasing of access to 'assortments', which is what a flat fee for so many statements interpreted would amount to, providing the 'assorted' dialogware is available.

One of the basic functions in marketing is 'assortment', putting together a complementary set of tools and materials for the consumer. This suggests a solution to the quality-use-time reporting dilemma. If the user pays a fixed price for a large-ish block of usage time with access to an assortment of dialog materials, then he or she will have



no scruples about rating individual chunks of interaction knowing that it won't affect the overall price unfavorably, and knowing, moreover, that it will probably enhance the quality of programs available in the future. If the assortment involves programs or routines written by different authors, then they can be paid varying royalties depending on the usage and ratings of their segments, (within limits). Each dialog, or independently authored routine, would require a unique name which would be used to record usage and ratings in a secure memory. When a certain volume of usage has accumulated, the system must shut off until a payment is made for another block of time with the assortment of application dialog programs.

The source code of the dialog programs can be made available in any convenient form; even broadcast by satellite super-stations at 300 baud FSK which can record on their home tape recorders. What needs to be controlled is the operation of the interpreter and the recording of usage time and quality. This leads one to envisage a sealed package containing a microprocessor, and ROM and RAM memory which would function something like a rechargeable battery.

#### 6. Implications of the scheme

Most people will soon be using computers for both business and amusement. A secure system which enables everyone to author programs, publish them, and collect feedback in terms of usage figures, ratings, and royalties, is absolutely

essential to avoid the kind of abortive collapse through piracy which now threatens the audio-recording industry, and, the publishing industry. The video disc looks nice to the large producers, but it won't be possible for people to record their own cheaply in the near future; conversely, the video-disc output can easily be copied onto a videocassette. Even it gives only an illusion of security. Security could be obtained via telecommunications and total encryption, but this would be terribly wasteful of communications channel capacity. Moreover, the end user could still make copies of some of the material. (Whatever does not require dynamic interaction.)

A sealed memory package for each program is all right if the program has no defects and is not too large. Many dialog programs are far too large for current ROMs. There is also the disadvantage that an expensive hardware object with unique characteristics is needed for each copy of the program. This leads to the kinds of inventory overhead problems which now plague publishers. The only hardware / object inventory involved in the "Go-be batts" system is that of the identical interpreter firmware batteries. Program source code can be stored in one place and transmitted by telecommunications, or printed bar codes or whatever as needed.

The main novel requirement is for clearinghouses to keep a record of validated authors, and to maintain audited use and rating records so that appropriate royalties etc. can be paid. Since the Post Office already recharges postage meters,

perhaps it ought to be the agency involved; also, it is probably above vested interests in application packages in a way that private corporations are unlikely to be. Finally, the Post Office has secure and private communications networks which would enable it to universalise the service easily. On the other hand, an altogether new agency might be more flexible and more capable of rapid development.

Again, as with postage meters, private firms could supply the "Go-be batts", perhaps leaving the interpreter, ROM, and encryption gate partly unwritten, to be completed by the clearinghouse agency at the time of first charging.

The fact that at present one software house is able to sell LISP programs cheaply in source code because it sells its micro-LISP interpreter for a higher price and then only in object code form, indicates that the general approach is practicable.

## 7. Summary

Information lacks the essential integrity to be a satisfactory commodity. But the services of a decision-maker go-between are readily marketed. The basic idea here is to sell the services of a go-between as the means for disseminating Computer Assisted Learning courseware and similar dialogware.



- Boyd, Gary M., "Contribution and Appraisal Networking as an Educational Technology", in Hills. P. and Gilbert, J., Aspects of Educational Technology XI, Kogan Page, London, pp. 203-208
- Brule, John D., "Realisation of a Public Key Cryptosystem", Creative Computing, vol. 5,7, July 1979, pp. 55-59
- Chapman, David, "Genie (An Immodest Proposal)", Dr. Dobbs' Journal of Computer Calisthenics and Orthodontia, vol. 4,2, February 1979, pp. 20-28
- Churchman, David, The Design of Inquiring Systems, Basic Books, New York 1971
- Denning, Dorothy S., "Secure Personal Computing in an Insecure Network", Communications of the ACM, vol. 22,8, August 1979, pp. 476-482
- National Research Council of Canada, NATAL 74 Formal Specifications, NRC, Ottawa, 1975
- Pask, Gordon, Conversation, Cognition, and Learning, Elsevier, 1975
- Queyssac, Daniel, "Projecting VLSI's Impact on Microprocessors", IEEE Spectrum, vol. 16,5, May 1979, pp. 38-40